Springer Series in Advanced Microelectronics 51

Ayan Palchaudhuri
Rajat Subhra Chakraborty

High Performance Integer Arithmetic Circuit Design on FPGA

Architecture, Implementation and Design Automation

Springer

Springer Series in Advanced Microelectronics

Volume 51

The Springer Series in Advanced Microelectronics provides systematic information on all the topics relevant for the design, processing, and manufacturing of microelectronic devices. The books, each prepared by leading researchers or engineers in their fields, cover the basic and advanced aspects of topics such as wafer processing, materials, device design, device technologies, circuit design, VLSI implementation, and subsystem technology. The series forms a bridge between physics and engineering and the volumes will appeal to practicing engineers as well as research scientists.

More information about this series at http://www.springer.com/series/4076

Ayan Palchaudhuri · Rajat Subhra Chakraborty

High Performance Integer Arithmetic Circuit Design on FPGA

Architecture, Implementation and Design Automation

Springer

Ayan Palchaudhuri
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology Kharagpur
Kharagpur, West Bengal
India

Rajat Subhra Chakraborty
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur, West Bengal
India

ISSN 1437-0387 ISSN 2197-6643 (electronic)
Springer Series in Advanced Microelectronics
ISBN 978-81-322-2519-5 ISBN 978-81-322-2520-1 (eBook)
DOI 10.1007/978-81-322-2520-1

Library of Congress Control Number: 2015943832

Springer New Delhi Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer (India) Pvt. Ltd. is part of Springer Science+Business Media (www.springer.com)

To Ma, for her continuous prayers,
support and inspiration.

Ayan Palchaudhuri

To Bunto, for her patience, love
and understanding.

Rajat Subhra Chakraborty

Preface

Realization of high performance arithmetic circuits targeted towards a specific family of the high-end Field Programmable Gate Arrays (FPGAs) continue to remain a challenging problem. Many fast arithmetic circuits proposed over the decades may not be amenable to efficient realization on a selected FPGA architecture. Experience has shown that current CAD tools for FPGAs are often unable to infer the native architectural components efficiently from the given input Hardware Description Language (HDL) specification of the circuit, as they explore only a small design space close to the input architectural description. The logic synthesis techniques inherent to the CAD tools are also often unable to apply the proper Boolean identities and perform appropriate algebraic factoring and sub-expression sharing, especially when intermediate signals are tapped out or registered. *Primitive instantiation* is an effective approach for optimization of designs on the Xilinx FPGA platform, and is often simpler than rewriting the Register Transfer Level (RTL) code to coax the logic synthesis tool to infer the desired architectural components. In addition, the FPGA CAD tools often fail to achieve an efficient placement of logic blocks on the FPGA fabric, resulting in higher routing delays.

In this book, we describe the optimized implementations of several arithmetic datapath, controlpath, and pseudorandom sequence generator circuits. We explore regular, modular, cascadable, and bit-sliced architectures for these circuits, by directly instantiating the target FPGA-specific primitives in the HDL specifications of the circuits. We justify every proposed architecture with detailed mathematical analyses. We improve performance by enforcing a constrained placement of the circuit building blocks, by placing the logically related hardware primitives in close proximity to one another, thereby minimizing the routing delay. This is accomplished by supplying relevant placement constraints in the Xilinx proprietary "User Constraints File" (.ucf) format to the FPGA CAD tool.

Taking advantage of the regularity of the architectures of the circuits proposed by us, the HDL specifications of the circuits as well as the placement constraints can be automatically generated. We have implemented a GUI-based CAD tool named *FlexiCore* integrated with the *Xilinx ISE* (Integrated Software Environment)

design environment for design automation of platform-specific high performance arithmetic circuits from user-level specifications. This tool was used to implement the proposed circuits, as well as hardware implementations of two integer arithmetic algorithms (Greatest Common Divisor (GCD) using Binary GCD algorithm and matrix multiplication using Distributed Arithmetic (DA)) where several of the proposed circuits were used as building blocks. Implementation results demonstrate higher performance and superior operand-width scalability at acceptable power-delay product (PDP) for the proposed circuits, with respect to implementations derived through other existing approaches.

Kharagpur

Ayan Palchaudhuri
Rajat Subhra Chakraborty

Acknowledgments

The authors thank Prof. Anindya Sundar Dhar, Department of Electronics and Electrical Communication Engineering, IIT Kharagpur, and Dr. Debdeep Mukhopadhyay, Department of Computer Science and Engineering, IIT Kharagpur, for their valuable insights into the work. The authors also acknowledge two undergraduate students of the Department of Computer Science and Engineering, IIT Kharagpur, Mohammad Salman and Sreemukh Kardas, for their contributions in developing the proposed CAD tool, *FlexiCore*.

This research was funded by a start-up research grant provided by IIT Kharagpur to Rajat Subhra Chakraborty.

Contents

Acronyms

ASIC	Application-Specific Integrated Circuit
BIST	Built-In Self-Test
CA	Cellular Automata
CAA	Cellular Automata Array
CAD	Computer Aided Design
CDI	Configuration Data In
CDO	Configuration Data Out
CE	Clock Enable
CL	Combinational Logic
CLB	Configurable Logic Block
CORDIC	COordinate Rotation DIgital Computer
DA	Distributed Arithmetic
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
FloPoCo	Floating-Point Cores
FF	Flip-Flop
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
GCD	Greatest Common Divisor
GUI	Graphical User Interface
HDL	Hardware Description Language
HKMG	High-K Metal Gate
HPL	High Performance Low Power
IP	Intellectual Property
ISE	Integrated Software Environment
LFSR	Linear Feedback Shift Register
LNS	Logarithmic Number System
LUT	Look-Up Table
MACC	Multiply Accumulate